DNS (Domain Name Server)

Suresh Podeti

4 min read · Jun 5, 2023

Photo by Sangharsh Lohakare on Unsplash

What is DNS?

The **domain name system (DNS)** is the Internet's naming service that **maps human-friendly domain names to machine-readable IP addresses.**

Fig 1.0: DNS maps domain names to IP Addresses

Important details

Name server

DNS servers that respond to users' queries are called **name servers**

Resource Record

The DNS database stores domain name to IP address mappings in the form of resource records (RR)

There are different types of RRs.

1. NS — Domain name → Hostname (*kittychef.in → dns.kittychef.in)*
   Provides the hostname that is the authoritative DNS for a domain name
2. A — Hostname → IP address *(dns.kittychef.in → 104.18.2.120)*
3. CNAME — Hostname → Canonical name *(Domain redirection eg: www.oldsite.com → www.newsisite.com)*
4. MX — Hostname → Canonical name. Mail exchanger

DNS Hierarchy

DNS name servers are in a hierarchical form. The hierarchical structure allows DNS to be highly scalable because of its increasing size and query load. A tree-like

structure is used to manage the entire DNS database.

There are mainly **three types of servers in the DNS hierarchy**:

1. **Root-level name servers:** These servers receive requests from local servers. Root name servers maintain name servers based on top-level domain names, such as `.com`, `.edu`, `.us`, and so on.
2. **Top-level domain (TLD) name servers:** These servers hold the IP addresses of authoritative name servers.
3. **Authoritative name servers:** These are the organisation's DNS name servers that provide the IP addresses of the web or application servers.

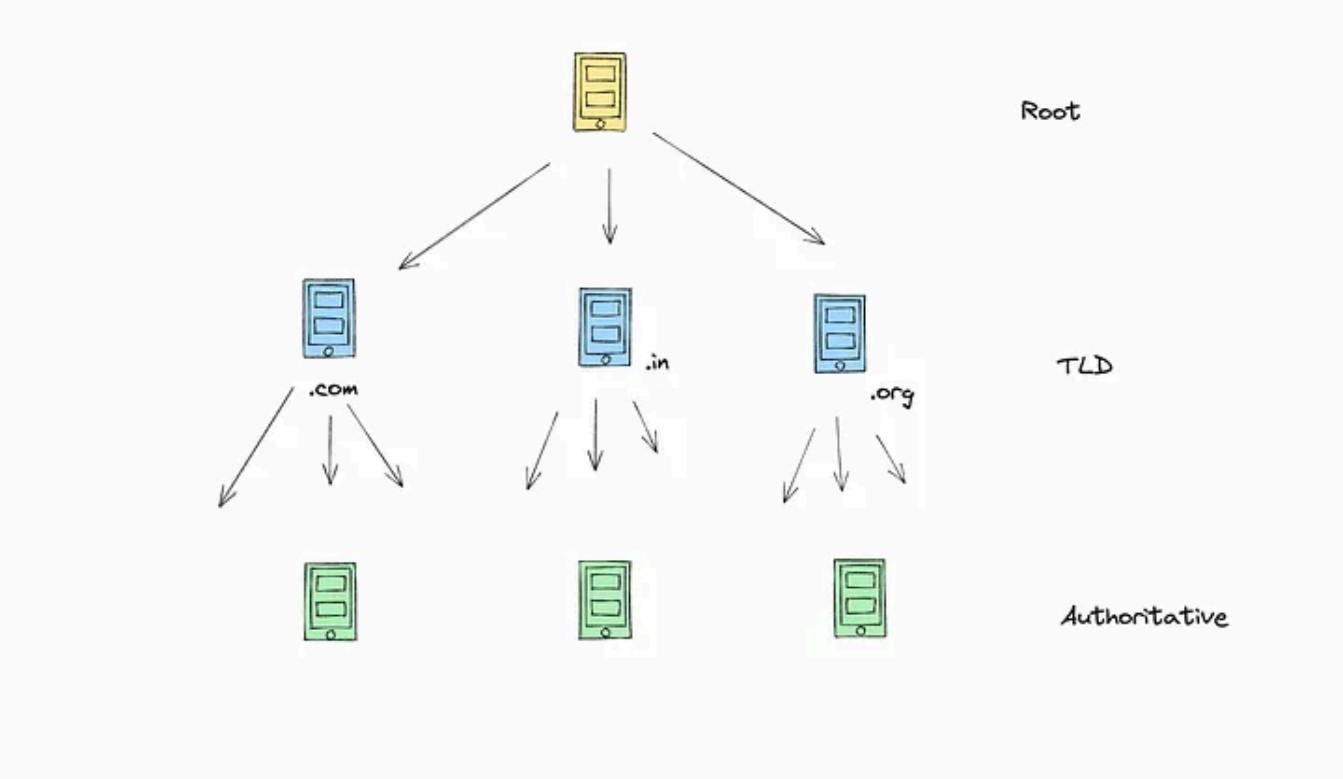

Fig 2.0: DNS hierarchy for resolution of domain/host names

Query resolution

There are two ways to perform a DNS query:

1. **Iterative**: The local server requests the root, TLD, and the authoritative servers for the IP address.
2. **Recursive**: The end user requests the local server. The local server further requests the root DNS name servers. The root name servers forward the requests to other name servers.

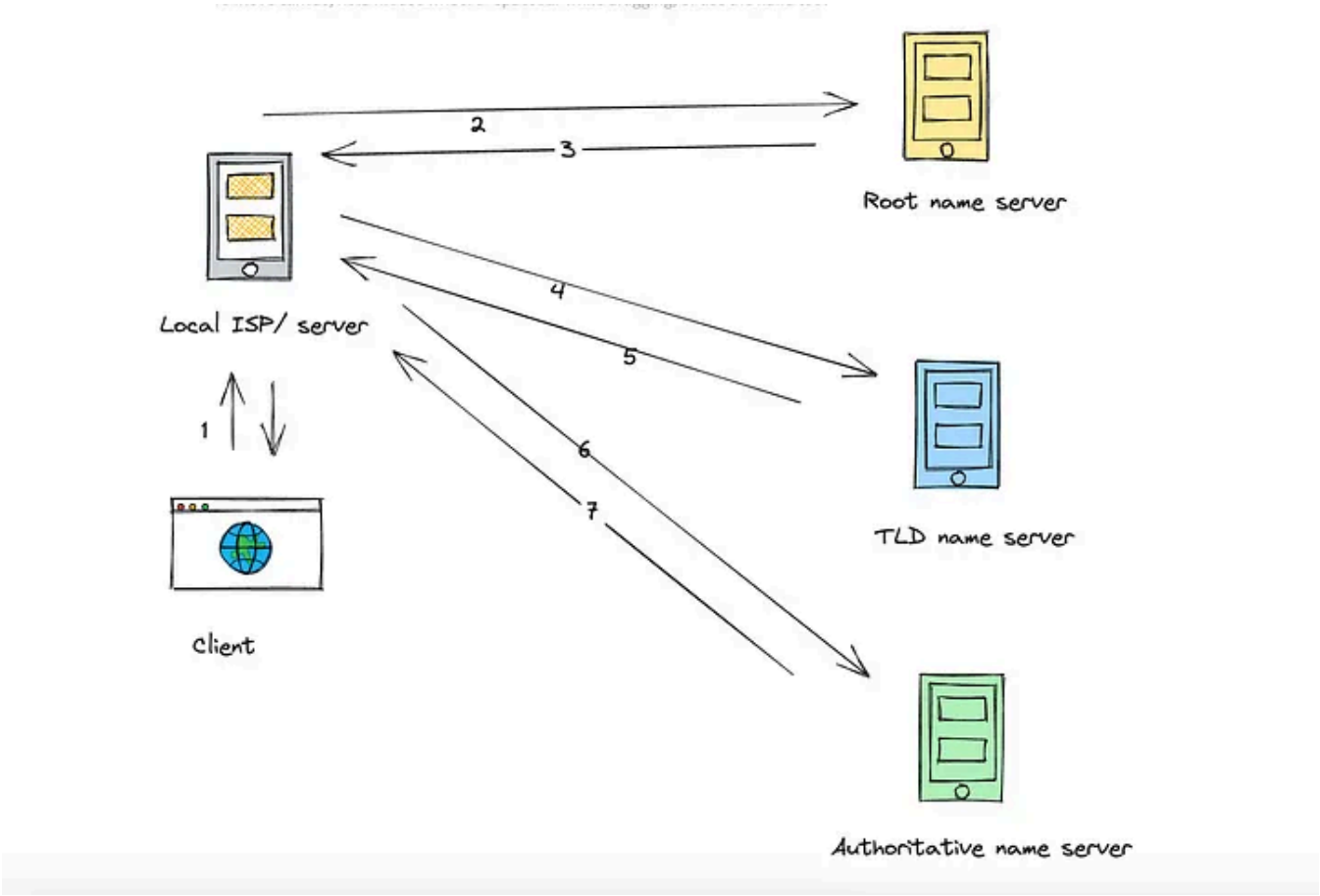

Fig 3.0: Iterative Query

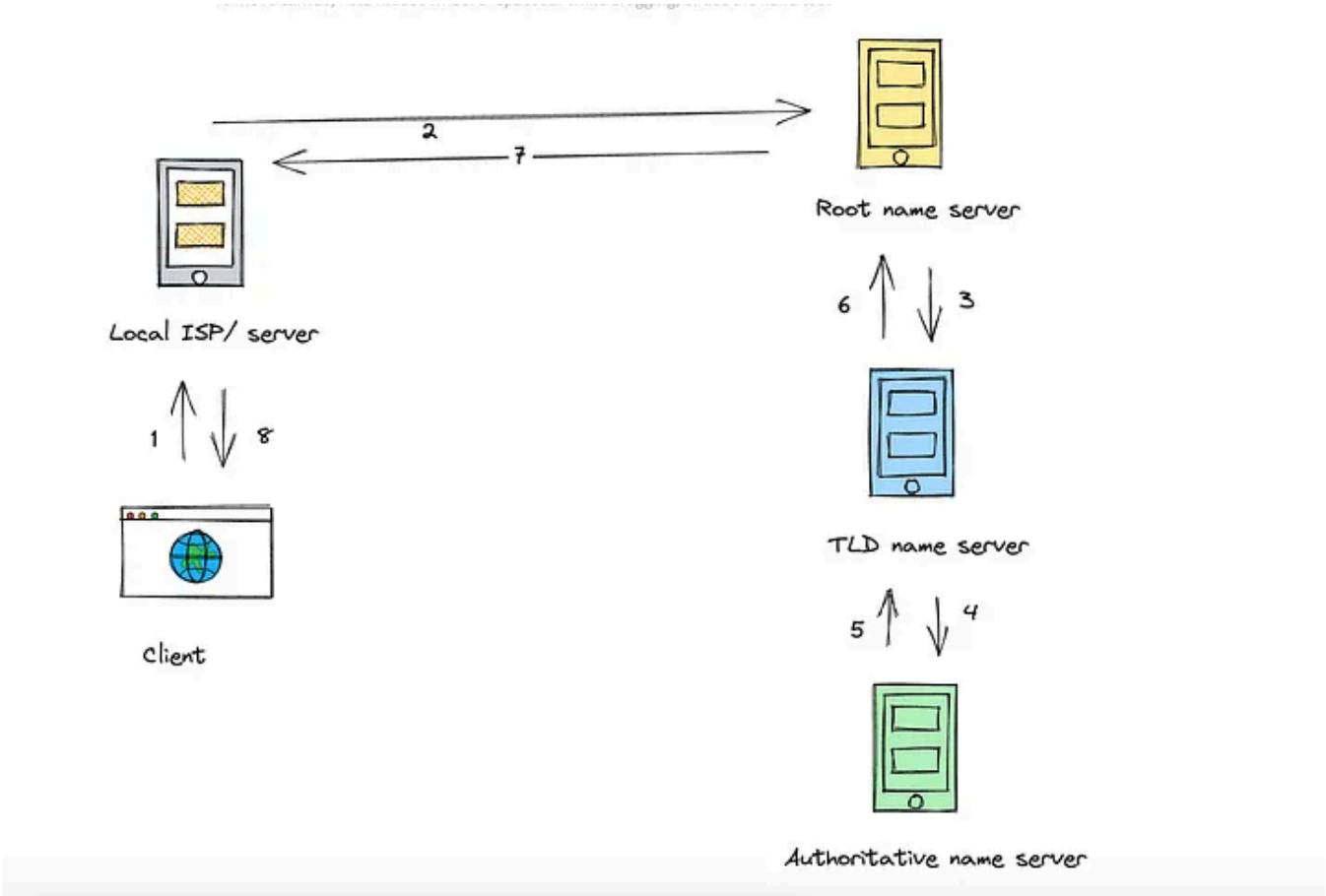

Fig 4.0: Recursive Query

> *Typically, an iterative query is preferred to reduce the query load on DNS infrastructure*

Caching

DNS uses caching at different layers to **reduce request latency** for the user. Caching plays an important role in reducing the burden on DNS infrastructure because it has to cater to the queries of the entire Internet.

DNS as a distributed system

Let's now go over how DNS is scalable, reliable, and consistent.

Highly scalable

Due to its hierarchical nature, DNS is a highly scalable system. Roughly 1,000 replicated instances of 13 root-level servers are spread throughout the world strategically to handle user queries. The working labor is divided among TLD and root servers to handle a query.

Reliable

Three main reasons make the DNS a reliable system:

1. **Caching:** The caching is done in the browser, the operating system, and the local name server, and the ISP DNS resolvers also maintain a rich cache of frequently visited services. Even if some DNS servers are temporarily down, cached records can be served to make DNS a reliable system.

2. **Server replication:** DNS has replicated copies of each logical server spread systematically across the globe to entertain user requests at low latency. The redundant servers improve the reliability of the overall system.

3. **Protocol:** Although many clients use DNS over unreliable user datagram protocol (UDP), UDP has its advantages. UDP is much faster and, therefore, improves DNS performance.

Consistent

DNS **compromises on strong consistency** to achieve high performance because data is read frequently from DNS databases as compared to writing. However, **DNS provides eventual consistency** and updates records on replicated servers lazily.

Consistency can suffer because of caching too, and each cached record comes with an expiration time called **time-to-live (TTL)** to mitigate this.

Test it out

Let's run a couple of commands. Click on the terminal to execute the following commands. Copy the following commands in the terminal to run them. Study the output of the commands:

1. `nslookup www.google.com`
2. `dig www.google.com`

Distributed Systems | System Design Interview | Software Architect | DNS | Software Architecture

Edit profile

Written by Suresh Podeti

1.2K Followers

Aspiring Software Architect | SDE III (Golang) @JungleeGames| Ex-Byju's | Co-founder @Rupant Tech. | Ex-Synopsys | I.I.T Kharagpur | 7+ Years of Experience

Recommended from Medium